JFA - 1

2025 - 2026

CM 2

CPU

Jean-François ANNE
jean-francois.anne@unicaen.fr
http://www.jfanne.fr
IUT de CAEN – Campus 3

IUT GRAND OUEST NORMANDIE
INFO
R 2.04

JFA 2

Sommaire

- Fonctionnement des processeurs d'un ordinateur
- Les composants du processeur
- Introduction au langage assembleur

IUT GRAND OUEST NORMANDIE
INFO
R 2.04

JFA - 3

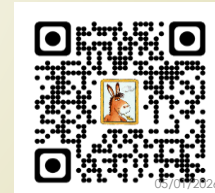
Fonctionnement des processeurs d'un ordinateur

DUT Informatique – Semestre 1

Ressource R2.04

Responsable : Jean-François ANNE

IUT GRAND OUEST NORMANDIE – INFO – R 2.04



JFA 4

Le processeur

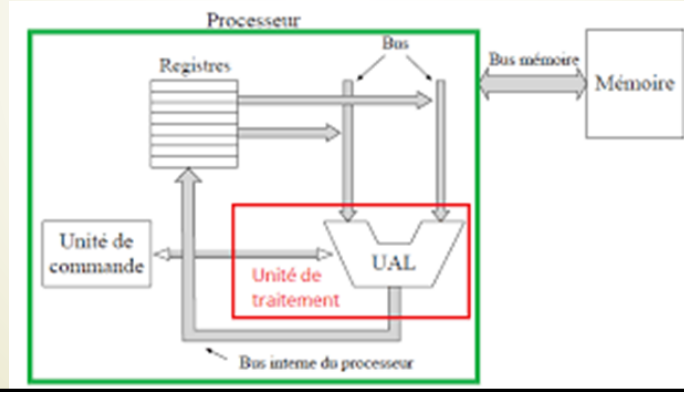
- Le processeur de l'ordinateur est la puce électronique la plus importante qui permet de faire fonctionner votre PC. C'est le « cœur » de l'ordinateur, en anglais on peut nommer ce dernier **CPU** pour central processing unit. Ce dernier contient des **cores ou cœurs physiques et logiques**.
- Enfin il existe beaucoup de type de processeurs car les fabricants Intel ou AMD proposent différentes gammes. Il est parfois difficile de s'y retrouver.
- Les instructions sont exécutées au niveau du CPU. Ce dernier va effectuer les calculs nécessaires pour que votre système d'exploitation et vos applications puissent fonctionner.

Le processeur

JFA 5

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

- Il est schématiquement constitué de 3 parties :
  - Les registres permettent de mémoriser de l'information (donnée ou instruction) au sein même du CPU. Leur nombre et leur taille sont variables en fonction du type de microprocesseur.
  - L'unité arithmétique et logique (UAL ou ALU en anglais) est chargée d'exécuter tous les calculs que peut réaliser le microprocesseur. On y retrouve des circuits comme l'additionneur, …
  - L'unité de commande permet d'exécuter les instructions (les programmes)



Le processeur

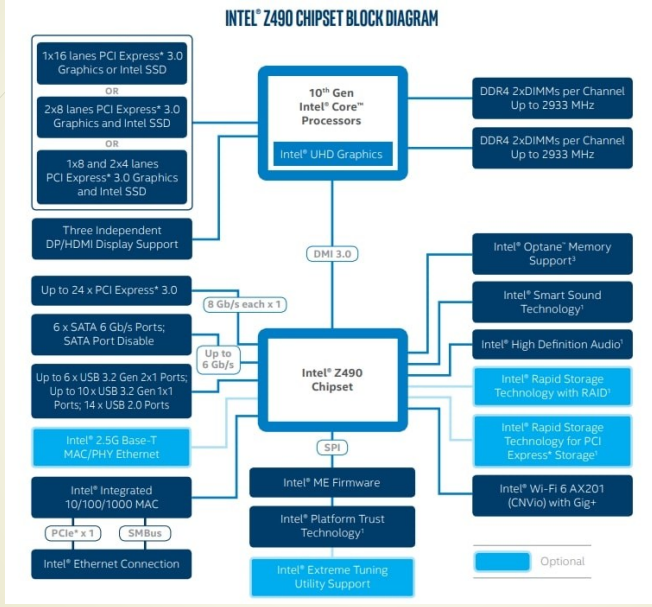
JFA 6

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

- D'un point de vue fonctionnel, le processeur effectue des calculs dont les données sont stockées dans la mémoire RAM de l'ordinateur. Des allés et venus se font entre ces deux composants pour que les logiciels s'exécutent.
- Plus les échanges sont rapides, plus le processeur va pouvoir effectuer de calculs. Ainsi le processeur embarque aussi un cache interne pour éviter de piocher les données redondantes dans la RAM. Cela accélère encore les traitements.
- Du côté de l'affichage et rendu graphique vers l'écran et le moniteur, les échanges se font entre le processeur et la carte graphique.
- La communication avec les périphériques plus lent se fait à travers le chipset.

Le processeur

JFA 7



il ne faut pas confondre CPU et GPU !

Le processeur

JFA 8

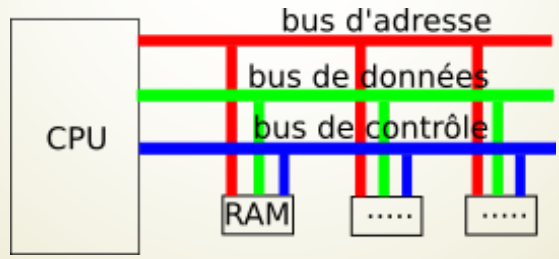
- D'un point de vue fonctionnel, le processeur effectue des calculs dont les données sont stockées dans la mémoire RAM de l'ordinateur. Des allés et venus se font entre ces deux composants pour que les logiciels s'exécutent.
- Plus les échanges sont rapides, plus le processeur va pouvoir effectuer de calculs. Ainsi le processeur embarque aussi un cache interne pour éviter de piocher les données redondantes dans la RAM. Cela accélère encore les traitements.
- Du côté de l'affichage et rendu graphique vers l'écran et le moniteur, les échanges se font entre le processeur et la carte graphique.
- La communication avec les périphériques plus lent se fait à travers le chipset.

JFA 9

Le Bus

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

- Les données doivent circuler entre les différentes parties d'un ordinateur, notamment entre la mémoire vive et le CPU. Le système permettant cette circulation est appelé bus. Il existe, sans entrer dans les détails, 3 grands types de bus :
  - Le bus d'adresse permet de faire circuler des adresses (par exemple l'adresse d'une donnée à aller chercher en mémoire)
  - Le bus de données permet de faire circuler des données
  - Le bus de contrôle permet de spécifier le type d'action (exemples : écriture d'une donnée en mémoire, lecture d'une donnée en mémoire).

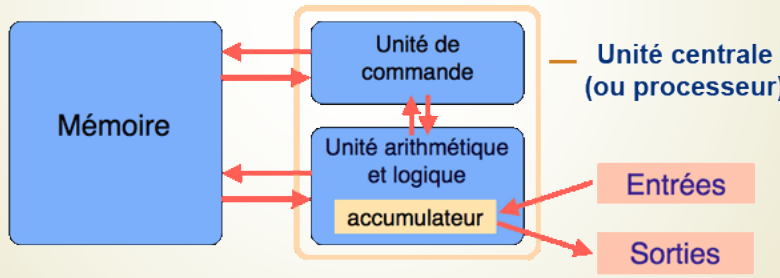


JFA 10

Architecture du CPU

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

- Les données et les instructions sont stockées en mémoire vive, les données et les instructions se partagent la mémoire vive (il n'y a pas une mémoire pour les instructions et une mémoire différente pour les données). C'est John von Neumann (mathématicien et physicien américano-hongrois 1903-1957) qui a eu l'idée en 1945 d'utiliser une structure de stockage unique pour les données et les instructions, voilà pourquoi on parle d'architecture de von Neumann. Voici un schéma qui représente ce modèle de von Neumann :



Modèle de von Neumann d'après le site https://interstices.info/

JFA 11

Architecture du CPU

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

- Sur ce schéma, nous avons :
  - La mémoire qui correspond à la RAM (Random Access Memory),
  - L'unité arithmétique et logique qui correspond à l'UAL (l'accumulateur est un registre permettant de stocker les résultats intermédiaires lors d'un calcul)
  - L'unité de commande qui gère l'exécution des instructions machines. À noter que cette unité de commande est aussi parfois appelée "unité de contrôle"
  - Le système "entrées-sorties" qui permet de communiquer avec "le monde extérieur" au système CPU+RAM (clavier, souris, écran, carte graphique, disque dur...)
- Pendant des années, pour augmenter les performances des ordinateurs, les constructeurs augmentaient la fréquence d'horloge des microprocesseurs : elle est liée à sa capacité d'exécuter un nombre plus ou moins important d'instructions machines par seconde. Plus la fréquence d'horloge du CPU est élevée, plus ce CPU est capable d'exécuter un grand nombre d'instructions machines par seconde (en simplifié). L'augmentation des fréquences a atteint la limite aux environs de 4 GHz. La seule façon d'augmenter encore les performances est maintenant de multiplier les cœurs sur la même puce.

JFA 12

Le jeu d'instruction

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

- Enfin un processeur se caractérise par son jeu d'instruction (addition, multiplications, ...) mais aussi de la manière dont les données sont stockées en mémoire.
- On distingue notamment les jeux d'instructions 32-bits et 64-bits.
- On trouve différents types de jeux d'instructions comme :
  - x86 qui sont les processeurs compatibles Intel 8086. On distingue x86-32 et x86-64 selon l'architecture 32-bits et 64-bits.
  - AMD64
  - PowerPC
  - SPARC
  - ARM
- Les logiciels et applications doivent être compilés spécifiquement pour fonctionner sur ce type d'instructions et ne peuvent donc fonctionner que pour ce type de processeur.
- Cela inclut aussi le système d'exploitation qui fonctionne avec un jeu d'instructions de processeur en particulier.

JFA 13

Fondeur de processeur

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

- Actuellement sur les PC, il existe deux constructeurs de processeurs principaux : AMD et Intel.

Chacun a ses propres gammes de processeurs et technologies associées qui sont liées à des appellations et des générations différentes.

1. Intel :

Par exemple, chez Intel on trouve des processeurs

- Processeurs Intel Pentium Gold et Silver : Processeurs d'entrée de gammes.
- Celéron ou Core m3 : Processeurs très basses consommations d'énergie pour ordinateur de type netbook
- i3 : Processeur de milieu de gamme et plutôt à destination des ordinateurs portables car conçu pour l'économie d'énergie.
- i5 : Processeur pour ordinateur de milieu et haut de gamme
- i7 : Processeur haut de gamme.
- i9 : sortie en 2017, c'est le très haut de gamme.

Chaque génération a ensuite ses sous-catégories, par exemple i9 se décline en i9 core et i9 Core Série X.

JFA 14

Fondeur de processeur

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

2. AMD

- Du côté de chez AMD, on trouve la génération Ryzen avec un numéro de version (et des modèles de type 1300X à 1950.

  Par exemple Ryzen 7 1800X. Plus le chiffre est élevé plus le processeur est haut de gamme.
- Enfin les processeurs ont des formats différents : l'emplacement sur la carte mère sur lequel il vient s'insérer est le **socket**.
- Ainsi, le processeur AMD ou Intel selon la génération ne peut fonctionner que sur certains types de cartes mères.
- Donc le choix de la carte mère impose le choix du processeur, et inversement !

Les composants du processeur

JFA 15

R 2.04

1. *L'horloge*

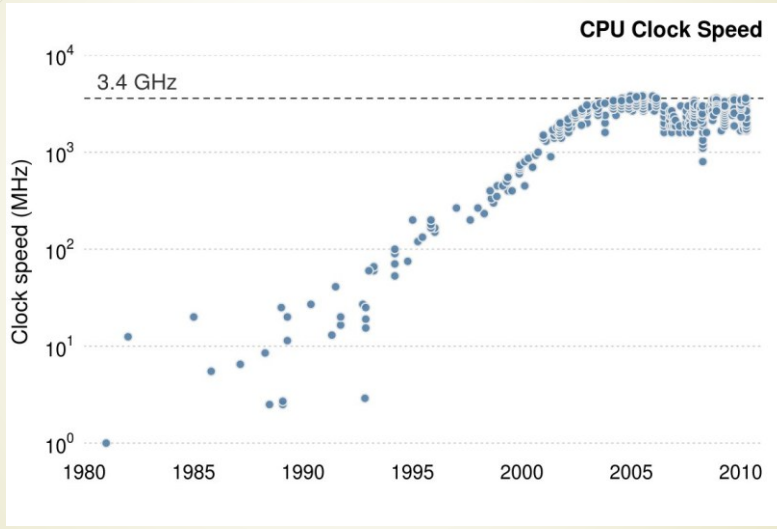
- **L'horloge du processeur** qui fournit un signal électrique régulier pour synchroniser ce dernier. Cette fréquence est exprimée en GHz, par exemple on dit un processeur de 3 GHz qui correspond à 3 trois milliards d'opérations en une seconde. Plus cette fréquence sera élevée et plus le processeur pourra effectuer d'opération.
- Le CPU est doté d'une **fréquence de base (BCLK)**, en général de 100 MHz à 200 MHz, fixé par la carte mère.
- A cette fréquence de base, on applique **un coefficient multiplicateur CPU**.
- Par exemple, un coefficient multiplicateur CPU de 46 et une horloge de base de 100 MHz donnent une vitesse d'horloge de 4,6 GHz.
- Le procédé consistant à augmenter la fréquence d'horloge se nomme **l'overclocking (surcadençage)**.

Les composants du processeur

JFA 16

R 2.04

1. *L'horloge*



https://ambssk.se/cpu-clock-speed-k.html

JFA 17

Les composants du processeur

R 2.04

1. *L'horloge*

- Comme vous pouvez le remarquer sur le graphique ci-dessus, à partir de 2006 environ, la fréquence d'horloge a cessé d'augmenter, pourquoi ?
- À cause d'une contrainte physique : en effet plus on augmente la fréquence d'horloge d'un CPU, plus ce dernier chauffe. Il devenait difficile de refroidir le CPU, les constructeurs de microprocesseurs ont décidé d'arrêter la course à l'augmentation de la fréquence d'horloge.
- Ils ont décidé d'adopter une nouvelle tactique. Comme il n'est plus vraiment possible d'augmenter les performances en augmentant la fréquence d'horloge des CPU, et bien augmentons le nombre de cœurs présent sur un CPU !

JFA 18

Le processeur multi-coeur

R 2.04

2. *Les cores ou coeurs*

- Dans un microprocesseur, un cœur est principalement composé : d'une UAL, de registres (R0, R1...) et d'une unité de commande,
- Un cœur est donc capable d'exécuter des programmes de façon autonome.
- La technologie permettant de graver plus de transistors, il est donc possible, sur une même puce, d'avoir plusieurs cœurs. Il s'agit de mini-processeurs qui peuvent effectuer des opérations de calculs en même temps. Le but est de diviser la file d'attente dans ces cœurs pour effecteur plusieurs opérations de calculs en même temps et donc multiplier la vitesse d'exécution. Aujourd'hui, on trouve sur le marché des CPU possédant jusqu'à 18 cœurs !
- On pourrait se dire que l'augmentation du nombre de cœurs entraîne obligatoirement une augmentation des performances du CPU, en fait, c'est plus que complexe que cela : pour une application qui n'aura pas été conçue pour fonctionner avec un microprocesseur multicœur, le gain de performance sera très faible, voir même nul. En effet, la conception d'applications capables de tirer profit d'un CPU multicœur demande la mise en place de certaines techniques de programmation (Programmation Parallèle).
- On parle alors de processeurs multi-cores ou multi-cœurs contrairement au mono-cœur. Par exemple le Ryzen 9 possède 12 cœurs.

JFA 19

Le processeur multi-coeur

R 2.04

3. *Les cores physiques*

- Un processeur n'est capable que d'effectuer une seule tâche à la fois en gérant une seule file d'attente (thread). Pour accélérer le traitement des instructions, on tente alors de faire exécuter plusieurs instructions en parallèle grâce à des cœurs logiques et physiques.
- Les cœurs physiques du processeur sont un découpage en plusieurs parties du processeur. En effet, la file d'attente des instructions à exécuter est répartie sur chacun des cœurs.
- Il faut aussi comprendre que la vitesse nominale du processeur est divisée par le nombre de cœurs.
  - Un processeur à 3.60 GHz avec 8 cœurs, aura une vitesse pour une tâche de : 3,60/8= 450 Mhz.
  - Si vous avez un processeur 4 cœurs physiques cadencés à 2,5Ghz, vous aurez en réalité 2,5/4 = 625 Mhz.
- Ainsi, le processeur à 2,5 Ghz sera plus rapide pour accomplir une seule tâche mais moins vite pour accomplir plusieurs tâches en même temps !
- Lorsqu'un processeur possède plusieurs cœurs, on parle de Processeur multi-cœurs (Multi-core processor).

JFA 20

Les 4 cœurs d'un processeur AMD

R 2.04

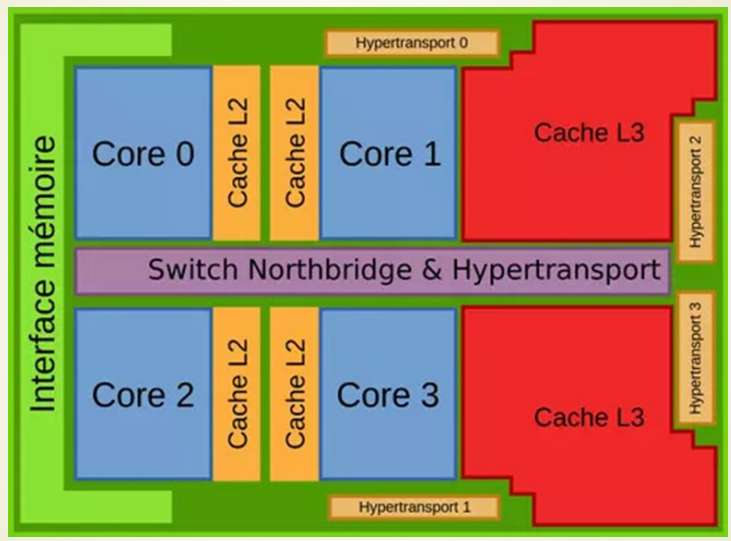


Schéma d'un AMD Phenom II quadro-cœurs

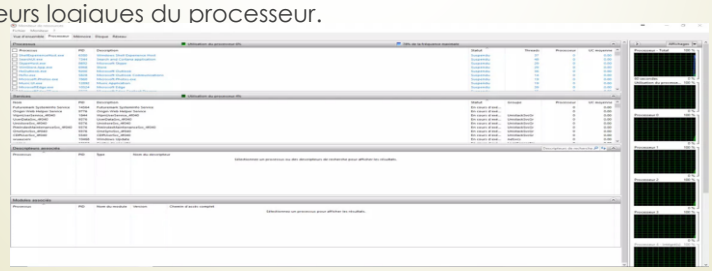
JFA 21

Le processeur multi-coeur

IUT GRAND OUEST NORMANDIE – iNFO – R 2.04

4. *Les cores logiques*

- Les cœurs logiques sont un découpage d'un cœur physique en deux par la technologie SM (HyperThreading chez Intel et AMD SMT).
- Ce dernier pourra alors gérer deux files d'attente au lieu d'une seule.
- Le but est d'utiliser un cœur à son maximum en s'assurant que ce dernier est toujours occupé. Il s'agit donc d'une optimisation de la file d'attente (thread).
- Comme un cœur physique possède deux cœurs logiques, il est assez facile de savoir qu'un processeur 4 cœurs aura 8 cœurs logiques.
- Une capture d'écran du moniteur de ressources systèmes où on voit à droite tous les cœurs logiques du processeur.



JFA 22

Le processeur multi-coeur

IUT GRAND OUEST NORMANDIE – iNFO – R 2.04

5. *La mémoire tampon ou cache*

- La mémoire cache ou tampon est une mémoire cache interne au processeur. Le but est d'être moins dépendant de la mémoire RAM afin d'éviter les échanges.
- Les différents cœurs d'un CPU doivent se "partager" l'accès à la mémoire vive : quand un cœur travaille sur une certaine zone de la RAM, cette même zone n'est pas accessible aux autres cœurs, ce qui va brider les performances.
- On trouve alors à l'intérieur des processeurs de la mémoire "ultrarapide" appelée mémoire cache. On peut stocker certaines données dans cette mémoire cache afin de pouvoir y accéder très rapidement dans le futur. En effet, l'accès à la mémoire cache est beaucoup plus rapide que l'accès à la RAM. La mémoire cache ayant un coût important, la quantité présente au sein d'un CPU est assez limitée. Les différents cœurs vont donc devoir se partager une partie de cette mémoire cache, ce qui peut aussi provoquer des ralentissements.

JFA 23

Le processeur multi-coeur

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

6. *Les niveaux de mémoire cache*

Il existe plusieurs niveaux de mémoires caches :

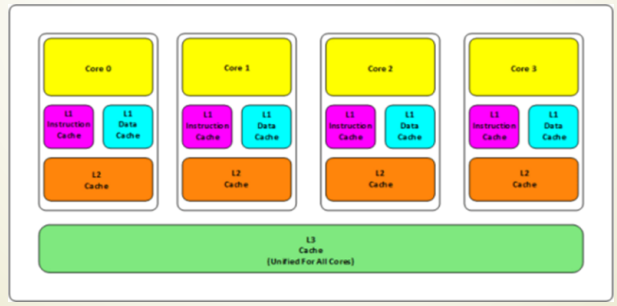
- **La mémoire cache de premier niveau**, appelée Cache L1 (Level 1 Cache) est directement intégrée dans le processeur. Les caches du premier niveau sont très rapides d'accès. Leur délai d'accès tend à s'approcher de celui des registres internes aux processeurs. Elle se subdivise en 2 parties :
  - Le cache d'instructions, qui contient les instructions issues de la mémoire vive décodées lors de passage dans les pipelines.
  - Le cache de données, qui contient des données issues de la mémoire vive et les données récemment utilisées lors des opérations du processeur.
- **La mémoire cache de second niveau**, appelée Cache L2 (Level 2 Cache) est située au niveau du boîtier contenant le processeur (dans la puce). Le cache de second niveau vient s'intercaler entre le processeur avec son cache interne et la mémoire vive. Il est plus rapide d'accès que cette dernière mais moins rapide que le cache de premier niveau. Elle contient les données qui ne sont pas dans le cache L1.

JFA 24

Le processeur multi-coeur

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

- **La mémoire cache de troisième niveau**, appelée Cache L3 (Level 3 Cache) est située dans le processeur ou à l'extérieur. Le cache de troisième niveau vient s'intercaler entre le processeur et la mémoire vive. Elle est plus lente d'accès que le cache L2 et le cache L1 mais sa capacité est plus grande. Elle contient les données qui ne sont pas dans le cache L1 et le cache L2.
- Chaque cœur du CPU a sa propre mémoire cache L1 pour les données et pour les instructions, tandis que chaque core du CPU a sa propre mémoire cache L2. Cependant, tous les cœurs de la CPU partagent la même mémoire cache L3. C'est une autre différence entre les caches L1 L2 et L3.



https://www.gushiciku.cn/pl/pJfh/zh-tw

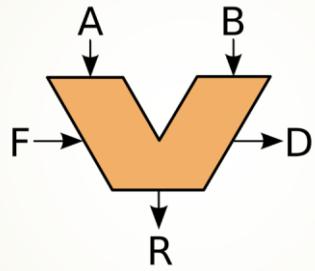
JFA 25

UAL du processeur

R 2.04

1.ALU (arithmetic logic unit) ou UAL (l'unité de calcul arithmétique et logique)

- c'est l'unité de calcul du processeur. Elle effectue des opérations arithmétiques sur des nombres entiers et logiques au niveau du bit.



R = (A) F (B)

- L'entrée A représente la première opérande, l'entrée B la deuxième opérande.
- L'entrée F permet de sélectionner l'opération que l'on veut effectuer.
- La sortie R contient le résultat de l'opération.
- La sortie D contient différents indicateurs sur le résultat de l'opération (zéro, retenue, négatif, débordement, …).

JFA 26

Les registres

R 2.04

- Les registres : ce sont des cases mémoires internes rapides qui peuvent stocker des informations intermédiaires ou utiles comme les données traitées par l'UAL, et l'adresse mémoire de l'instruction en cours d'exécution ou de la suivante (en fonction de l'architecture). Il y en a plusieurs dans un processeur, avec pour certains registres des utilisations particulières.

  - **L'accumulateur :**

    L'accumulateur est un registre particulier, qui est utilisé en liaison avec l'UAL.

  - **L'unité de contrôle ou de commande ou de traitement**

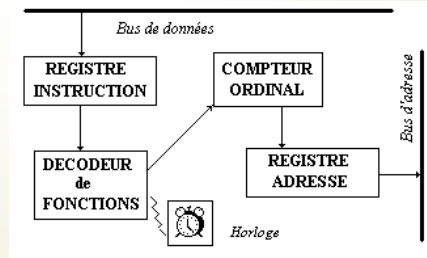
    L'Unité de contrôle contrôle le fonctionnement du processeur, notamment du chemin de données, rassemble les instructions en mémoire pour les envoyer à l'ALU, aux registres et aux autres composants. Elle est chargée de commander et de gérer tous les différents constituants le CPU (contrôler les échanges, gérer l'enchaînement des différentes instructions, etc...).

JFA 27

Les registres

R 2.04

- Elle est composée au minimum de :
  - d'un registre instruction RI,
  - d'un compteur ordinal CO,
  - d'un registre adresse RA,
  - d'un décodeur de fonctions,
  - d'une horloge.
- Le séquenceur de l'unité de contrôle gère le timing des opérations, les registres à utiliser, les interruptions, etc.



https://rmdiscala.developpez.com/cours/LesChapitres.html/Cours1/Chap1.5.htm

JFA 28

Le langage assembleur

R 2.04

- Le langage assembleur est dit de "bas niveau " car il est étroitement lié à l'architecture du microprocesseur.
- Les « instructions machines" sont exécutées par l'unité de commande. Les instructions exécutées au niveau du CPU sont donc codées en binaire. L'ensemble des instructions exécutables directement par le microprocesseur constitue ce que l'on appelle le "langage machine".
- Une instruction machine est une chaîne binaire composée principalement de 2 parties :
  - Le champ "code opération" qui indique au processeur le type de traitement à réaliser. Par exemple le code "00100110" donne l'ordre au CPU d'effectuer une multiplication.
  - Le champ "opérandes" indique la nature des données sur lesquelles l'opération désignée par le "code opération" doit être effectuée.

champ code opération	champ opérandes

Instruction machine

JFA 29

Le langage assembleur

R 2.04

- Les instructions machines sont relativement basiques (on parle d'instructions de bas niveau), voici quelques exemples :
  - Les instructions arithmétiques (addition, soustraction, multiplication...).
    - On peut avoir une instruction "additionne la valeur contenue dans le registre R1 et le nombre 789 et range le résultat dans le registre R0.
  - Les instructions de transfert de données qui permettent de transférer une donnée d'un registre du CPU vers la mémoire vive et vice versa.
    - On peut avoir une instruction "prendre la valeur située à l'adresse mémoire 487 et la placer dans le registre R2" ou encore "prendre la valeur située dans le registre R1 et la placer à l'adresse mémoire 512"
  - Les instructions de rupture de séquence d'exécution encore appelées instructions de saut ou de branchement permettent d'interrompre l'ordre initial sous certaines conditions en passant à une instruction située une adresse mémoire donnée
    - On peut avoir une instruction "si la valeur contenue dans le registre R1 est strictement supérieure à 0 alors exécuter l'instruction située à l'adresse mémoire 4521", dans le contraire, la prochaine instruction à exécuter est à l'adresse mémoire 355.

JFA 30

Le langage assembleur

R 2.04

- Il n'est pas question ici d'apprendre à programmer en assembleur, mais voici tout de même quelques instructions de base en assembleur :
  - **LDR R1,78 : (LoaD Register) :** Place la valeur stockée à l'adresse mémoire 78 dans le registre R1.
  - **STR R3,125 : (STore Register) :** Place la valeur stockée dans le registre R3 en mémoire vive à l'adresse 125.
  - **ADD R1, R0, #128 : (Addition) :** Additionne le nombre 128 (Le symbole # identifie une valeur immédiate) et la valeur stockée dans le registre R0, place le résultat dans le registre R1.
  - **SUB R0, R1, R2 : (Soustraction) :** Soustrait la valeur stockée dans le registre R2 de la valeur stockée dans le registre R1, place le résultat dans le registre R0.
  - **MOV R1, #23 : (déplacements entre registres)** Place le nombre 23 dans le registre R1.
  - **MOV R0, R3 :** Place la valeur stockée dans le registre R3 dans le registre R0.
  - **B 45 : (Branch to)** C'est une rupture de séquence, la prochaine instruction à exécuter se situe à l'adresse 45.

JFA 31

Le langage assembleur

R 2.04

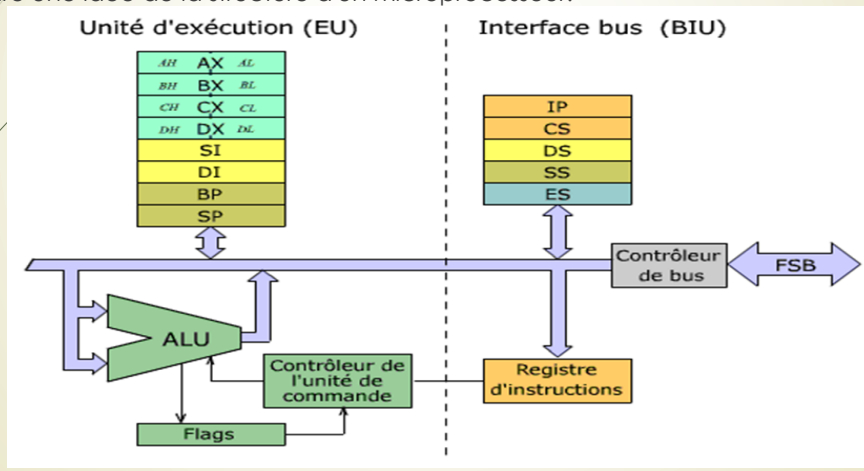
- **CMP R0, #23 :** Compare la valeur stockée dans le registre R0 et le nombre 23. Cette instruction CMP doit précéder une instruction de branchement conditionnel **BEQ, BNE, BGT, BLT .**
- **CMP R0, R1 :** Compare la valeur stockée dans le registre R0 et la valeur stockée dans le registre R1.
- **CMP R0, #23**
- **BEQ 78 :** La prochaine instruction à exécuter se situe à l'adresse mémoire 78 si la valeur stockée dans le registre R0 est égale à 23.
- **HALT :** Arrête l'exécution du programme.

JFA 32

Les registres 16 bits du processeur

R 2.04

- Les registres ci-dessous sont les registres du 8086, l'ancêtre des processeurs actuels. Ces processeurs ont évolué depuis tout en restant compatibles avec leur ancêtre (compatibilité ascendante). Cette description devrait donc suffire à se faire une idée de la structure d'un microprocesseur.

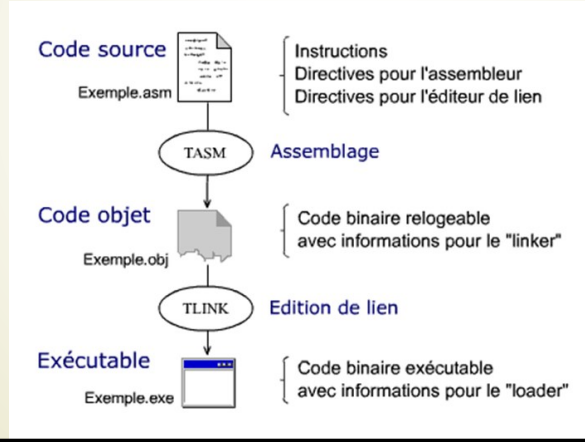


JFA 33

Assemblage et édition de liens

R 2.04

- Pour rappel, on part d'un code source ".asm". L'assemblage en fait un code objet ".obj" relogeable qui à son tour est traité par l'éditeur de lien pour en faire un code exécutable ".exe". Accessoirement l'assembleur produit aussi un "listing" qui est un document destiné à être imprimé. On y retrouve côte à côte, le code source et le code binaire correspondant noté en hexadécimal.



JFA 34

Exercices

R 2.04

- ***Exercice 1 :***

Expliquez brièvement, les instructions suivantes :

ADD R0, R1, #42

LDR R5,98

CMP R4, #18

BGT 77

STR R0,15

B 100

JFA 35

Corrigé des Exercices

IUT GRAND OUEST NORMANDIE – iNFO – R 2.04

- ***Exercice 1 :***

Expliquez brièvement, les instructions suivantes :

- **ADD R0, R1, #42**

  Additionne le nombre 42 et la valeur stockée dans le registre R1, place le résultat dans le registre R0.

- **LDR R5,98**

  Place la valeur stockée à l'adresse mémoire 98 dans le registre R5.

- **CMP R4, #18**

  BGT 77

  La prochaine instruction à exécuter se situe à l'adresse mémoire 77 si la valeur stockée dans le registre R4 est supérieure à 18.

- **STR R0,15**

  Place la valeur stockée dans le registre R0 en mémoire vive à l'adresse 15.

- **B 100**

  La prochaine instruction à exécuter se situe à l'adresse 100.

JFA 36

Exercices

IUT GRAND OUEST NORMANDIE – iNFO – R 2.04

- ***Exercice 2 :***

Écrire les instructions en assembleur correspondant aux phrases suivantes :

- Additionne la valeur stockée dans le registre R0 et la valeur stockée dans le registre R1, le résultat est stocké dans le registre R5
- Place la valeur stockée à l'adresse mémoire 878 dans le registre R0
- Place le contenu du registre R0 en mémoire vive à l'adresse 124
- La prochaine instruction à exécuter se situe en mémoire vive à l'adresse 478
- Si la valeur stockée dans le registre R0 est égale 42 alors la prochaine instruction à exécuter se situe à l'adresse mémoire 85

JFA 37

Corrigé des Exercices

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

Exercice 2 :

Écrire les instructions en assembleur correspondant aux phrases suivantes :

- Additionne la valeur stockée dans le registre R0 et la valeur stockée dans le registre R1, le résultat est stocké dans le registre R5.
  - **ADD R5, R1, R0**
- Place la valeur stockée à l'adresse mémoire 878 dans le registre R0.
  - **LDR R0, 878**
- Place le contenu du registre R0 en mémoire vive à l'adresse 124.
  - **STR R0,124**
- La prochaine instruction à exécuter se situe en mémoire vive à l'adresse 478.
  - **B 478**
- Si la valeur stockée dans le registre R0 est égale 42 alors la prochaine instruction à exécuter se situe à l'adresse mémoire 85.
  - **CMP R0, #42**
    BEQ 85

JFA 38

Exercices

IUT GRAND OUEST NORMANDIE – INFO – R 2.04

Exercice 3 :

Voici un programme Python :

```
x = 4
y = 8
if x == 10:
    y = 9
else :
    x=x+1
z=6
```

Après avoir analysé très attentivement le programme en assembleur ci-contre, établir une correspondance entre les lignes du programme en Python et les lignes du programme en assembleur.

- À quoi sert la ligne "B endif" ?
- À quoi correspondent les adresses mémoires 23, 75 et 30 ?

Et voici maintenant voici son équivalent en assembleur :

```
  MOV R0, #4
  STR R0,30
  MOV R0, #8
  STR R0,75
  LDR R0,30
  CMP R0, #10
  BNE else
  MOV R0, #9
  STR R0,75
  B endif
else:
  LDR R0,30
  ADD R0, R0, #1
  STR R0,30
endif:
  MOV R0, #6
  STR R0,23
  HALT
```

JFA 39

Corrigé des Exercices

R 2.04

- ***Exercice 3 :***

Voici un programme Python :

```
x = 4
y = 8
if x == 10:
    y = 9
else :
    x=x+1
z=6
```

Après avoir analysé très attentivement le programme en assembleur ci-contre, établir une correspondance entre les lignes du programme en Python et les lignes du programme en assembleur.

- À quoi sert la ligne "B endif" ?
- À quoi correspondent les adresses mémoires 23, 75 et 30 ?

Et voici maintenant voici son équivalent en assembleur :

```
   MOV R0, #4
   STR R0,30
   MOV R0, #8
   STR R0,75
   LDR R0,30
   CMP R0, #10
   BNE else
   MOV R0, #9
   STR R0,75
   B endif
else:
   LDR R0,30
   ADD R0, R0, #1
   STR R0,30
endif:
   MOV R0, #6
   STR R0,23
   HALT
```

- À quoi sert la ligne "B endif" ?
  - C'est une rupture de séquence, on continue le programme en ligne endif
- À quoi correspondent les adresses mémoires 23, 75 et 30 ?
  - X est à l'adresse mémoire 30.
  - Y est à l'adresse mémoire 75.
  - Z est à l'adresse mémoire 23.

JFA 40

WEBOGRAPHIE

R 2.04

- Liens Web :
- https://www.malekal.com/chipset-northbridge-pch-fch-role-definition/
- https://www.malekal.com/caracteristiques-fonctionnement-processeurs-ordinateur/
- https://books.google.fr/books?id=xLxsDwAAQBAJ&pg=PA195&lpg=PA195&dq=technologue+cpu+ual&source=bl&ots=WiDaRupT0r&sig=ACfU3U1O-HsFNHj4egrLVvyU-ZfZtFvg9A&hl=fr&sa=X&ved=2ahUKEwib54WIl4n1AhUFx4UKHXrxAS4Q6AF6BAgREAM#v=onepage&q=technologue%20cpu%20ual&f=false
- https://www.malekal.com/quest-ce-que-hyper-threading-pour-un-processeur-et-cpu/
- https://www.malekal.com/composants-fonctionnement-ordinateur/
- https://learntutorials.net/fr/x86/topic/2122/notions-fondamentales-sur-les-registres
- https://www.courstechinfo.be/Programmation/IntroASM.html#bas
- https://pixees.fr/informatiquelycee/n_site/nsi_prem_von_neu.html